Bike-Manufaktur

Aufgaben

In den letzten Jahren ist die Nachfrage nach Fahrrädern und E-Bikes stark gestiegen. Ein neu gegründetes Unternehmen, das hochwertige Bikes nach Kundenwünschen herstellt und mit dem Fachhandel zusammenarbeitet, fertigt die Rahmenkits (Rahmen und Rahmenbestandteile) selbst an. Die speziellen Elemente der Schaltgruppen und die Einzelteile werden von ausgewählten Herstellern zugeliefert und bei der Montage entsprechend angepasst. Für die Bike-Manufaktur sollen ein Softwaresystem zur Unterstützung des Herstellungsprozesses und eine Datenbank zur Verwaltung der montierten und ausgelieferten Bikes entwickelt werden. Außerdem soll für einen Bikeshop eine neue Datenbank zur Speicherung von Verkaufs- und Leasingdaten modelliert werden.

1 Objektorientierte Entwicklung für die Fertigungsorganisation der Bikes
Die Kundin oder der Kunde (im Folgenden Kunde genannt) wählen für Rahmenkit, Schaltgruppe und diverse Einzelteile aus dem Gesamtsortiment aus. Die Auswahl wird in einem Datenblatt festgehalten und führt zu einer Konfigurationsdatei für die Herstellung des gewünschten Bikes (Material 1). Bei der Montage, beispielsweise von einem Rennrad, beginnt man mit dem Rahmen und allen Teilen, die zum ausgewählten Rahmenkit gehören. Anschließend erfolgt der Einbau des Sattels (Einzelteil) und nach der Montage einiger Schaltelemente der Einbau des Lenkers (Einzelteil). Dann werden die restlichen Schalt-/Bremselemente sowie die Räder (Einzelteile) verbaut. Ein UML-Klassendiagramm finden Sie in Material 2, die Dokumentation der Klasse `String` in Material 3.

1.1 Die Klasse `Komponente` hat bei der Modellierung der Bestandteile eines Bikes eine besondere Bedeutung. Beschreiben Sie die im Klassendiagramm in Material 2 vorliegende Vererbungsstruktur (Stichworte: Spezialisierung, Generalisierung, abstrakte Klassen, Methoden, Zugriffsrecht `protected`, Polymorphie). Erläutern Sie die Beziehung zwischen den Komponenten und deren Bedeutung für die Implementierung einer doppelt verketteten Liste.

(6 BE)

1.2 Überführen Sie die Klassen `Komponente` und `RahmenTeil` (Material 2) in Anweisungen einer objektorientierten Programmiersprache und implementieren Sie die Konstruktoren und Methoden.

Hinweise: Der Parameter `part0` des Konstruktors der Klasse `Komponente` enthält alle Attributwerte, jeweils durch Komma getrennt. Analog zusammengesetzt ist der zweite Konstruktor-Parameter `part1` der Komponententypen.

(8 BE)

1.3 Im Material 1 sind exemplarisch Daten für die Montage eines Rennrads zusammengestellt. Entwickeln und zeichnen Sie ein UML-Objektdiagramm für das dargestellte Bike.

Hinweise: Für das Objektdiagramm sind aus der Tabelle in Material 1 nur die Komponenten von ID `r3` bis ID `s12` darzustellen. Beachten Sie auch das Klassendiagramm in Material 2. Nur für E-Bikes ist zwingend eine Versicherung notwendig. Auf die Darstellung des Attributs `String[] konfigzeilen` kann hier verzichtet werden.

(8 BE)

1.4 Beim Herstellen eines Bikes ist der Rahmen immer das erste Bauteil. Die Methode `hinzufuegenKomp()` soll eine automatisierte Montage eines Fahrrads in gegebener Einbaureihenfolge nachbilden. Die Demontage eines Bikes ist genau in der umgekehrten Reihenfolge durchzuführen, weshalb die Komponenten doppelt verkettet werden. Implementieren Sie die Klasse `Bike` mit ihrem Konstruktor sowie der Methode `hinzufuegenKomp()`.

Hinweise: Die Methode `hinzufuegenKomp()` fügt ab der zweiten Komponente eine neue Komponente hinzu. Wenn bereits alle Teile des Bikes montiert sind, gibt sie `false` zurück. Alle Komponenten des Bikes werden bei der Objekterzeugung automatisch hinzugefügt.

(12 BE)

1.5 Es kann erforderlich sein, dass eine in ein Bike eingebaute Komponente ersetzt werden muss. Für den Fall, dass man dafür Komponenten demontieren muss, ist die Kenntnis der Reihenfolge der Demontage vom letzten bis einschließlich zum zu ersetzenden Bauteil erforderlich. Die Methode `ersetzeKomp()` ersetzt die Komponente mit der gleichen ID und gibt einen String mit den Angaben der auszubauenden Komponenten in der richtigen Reihenfolge zurück. Falls keine Komponente ersetzt wird, soll der zurückgegebene String diese Information enthalten. Implementieren Sie die Methode `ersetzeKomp()`.

Hinweise: Die zu ersetzende Komponente wird anhand der mitgelieferten ID identifiziert. Da man die erste Komponente, den speziellen Rahmen, nicht ersetzen muss, ist davon auszugehen, dass das Bike mindestens aus zwei montierten Teilen besteht, wenn die Methode `ersetzeKomp()` ausgeführt wird.

(7 BE)

1.6 Für das montierte Bike wird in der Methode `berechnePreis()` der Gesamtpreis ermittelt, wobei es einige Besonderheiten zu beachten gilt.
Das Unternehmen will seine aktuellen Restbestände von Rahmenteilen der Gruppe Manu X abverkaufen und gewährt dafür einen Rabatt von 10% auf den Gesamtpreis.
Neuerdings bietet das Unternehmen seinen Kunden auch Bikes mit Rädern ohne Schlauch an, d.h. der Name eines montierten Einzelteils beginnt mit Tubeless. Dadurch ist die Montage wesentlich komplizierter und bedeutet für jede dieser Tubeless-Komponenten 12€ Zusatzkosten, die noch nicht im Einzelpreis enthalten sind.
Wegen stark erhöhter Nachfrage ist der Aufwand zur Beschaffung von qualitativ hochwertigen Scheibenbremsen gestiegen. Für jede Komponente mit der Bezeichnung Bremsscheibe werden deshalb zusätzlich 8€ berechnet.
Entwickeln und zeichnen Sie ein Struktogramm für diese Methode.

(9 BE)

1.7 Die Fertigungsorganisation soll durch ein Verwaltungssystem erweitert werden.
Folgende Funktionalitäten sollen vom System bereitgestellt werden:

- Kunden konfigurieren ihr zukünftiges Bike, das Servicepersonal kann sie dabei unterstützen.
- Das Servicepersonal ist zuständig für die Auslieferung des fertigen Bikes, was voraussetzt, dass das Bike von einer Technikerin bzw. einem Techniker (im folgenden Techniker genannt) montiert und der Preis vom Servicepersonal ermittelt worden ist.
- Im Montageprozess der Fahrräder wird regelmäßig ein Bike ausgewählt, das ausführlich im Testlabor geprüft wird. Diese Prüfung liegt in der Verantwortung eines Technikers. Sollte sich dabei zeigen, dass eine montierte Komponente des Bikes den hohen Belastungen nicht standhält, wird sie von einem Techniker ausgetauscht.

– Das Unternehmen bietet für seine Bikes auch Zubehör an, zum Beispiel Flaschenhalter und Beleuchtung. Wenn der Kunde Zubehör wünscht, wird die Auslieferung des Bikes um den Anbau von Zubehör erweitert. Dafür ist wiederum ein Techniker zuständig.

1.7.1 Beschreiben Sie den Zweck eines Anwendungsfalldiagramms und ordnen Sie seine Verwendung in den Softwarelebenszyklus ein.

(2 BE)

1.7.2 Entwickeln und zeichnen Sie für die unter Aufgabe 1.7 genannten Funktionalitäten ein Anwendungsfalldiagramm in UML-Notation.

(8 BE)

2 Entwicklung der Datenbank zur Verwaltung der montierten und ausgelieferten Bikes
Ein erstes Entity-Relationship-Modell (ERM) der Datenbank finden Sie in Material 4.

2.1 Geben Sie drei unterschiedliche Lösungen an, um die Spezialisierung zwischen `Komponente`, `Rahmenteil`, `Schaltelement` und `Einzelteil` in das relationale Modell zu transformieren. Diskutieren Sie die Vor- und Nachteile dieser Möglichkeiten.

(6 BE)

2.2 Überführen Sie das ERM (Material 4) unter Beachtung von Aufgabe 2.1 in das Relationenmodell in der 3. Normalform.

Hinweis: Alle Relationen sind in der Schreibweise `Relation(PK, Attribut, ..., FK#)` (PK unterstrichen) anzugeben.

(5 BE)

2.3 Einige Daten der Datenbank sollen ergänzt, geändert bzw. ausgewertet werden.

2.3.1 Die Komponenten des Herstellers Manu verteuern sich um 5%.
Formulieren Sie eine SQL-Anweisung zur Aktualisierung der Datenbank.

(2 BE)

2.3.2 Es wird die neue Kategorie E-Bike-Trekking mit notwendigem Versicherungsschutz in die Datenbank aufgenommen. Das nächste Bike, das fertiggestellt wird, ist ein Rad dieser Kategorie (Modell Manu SportABC). Für die Auslieferung an den Kunden wurde dafür ein neuer Fachhändler gewonnen: MyChain, 63452 Hanau, Einkaufsstraße 28.
Formulieren Sie die SQL-Anweisungen, um diese Daten einzutragen.

Hinweise: Die IDs der Tabellen `Fachhändler` und `Bike` werden vom System automatisch vergeben. Die Funktion `NOW()` liefert das aktuelle Tagesdatum, die Funktion `YEAR()` die zu einem Datum zugehörige Jahreszahl.

(6 BE)

2.3.3 Formulieren Sie eine SQL-Anweisung, um alle im letzten Jahr gebauten Bikes, die an den Fachhändler "Bosse Zweirad" in Köln geliefert wurden, mit Bike-ID und Bezeichnung von Modell und Kategorie auszugeben.

(3 BE)

2.3.4 Die BikeID und das Gesamtgewicht jedes Racing-Bikes sollen ausgegeben werden.
Implementieren Sie eine entsprechende SQL-Anweisung.

(3 BE)

2.3.5 Für alle Vorjahresmodelle will man diejenigen Hersteller ermitteln, von denen mindestens zehn Felgenbremsen (Detail „Felgen-Br“) verbaut worden sind.
Entwickeln Sie eine SQL-Anweisung, die diese Informationen liefert.

Hinweis: Die Angabe der Anzahl ist absteigend zu sortieren.

(5 BE)

3 Für einen Bikeshop soll eine neue Datenbank entwickelt werden, die folgende Anforderungen umsetzt:

- Von einer Herstellerin oder einem Hersteller (im Folgenden Hersteller genannt) werden zu einem bestimmten Datum und zu einem bestimmten Einkaufspreis Bikes geliefert. Für das gezielte Anbieten dieser Bikes sind die Eigenschaften Farbe, Rahmengröße, Laufradgröße, Schaltung (Gruppe), Produktzustand und Verfügbarkeit erforderlich.
- Hersteller zeichnen sich durch Namen und Ort aus.
- Jedes Bike kann durch unterschiedliches Zubehör ergänzt werden. Dies ist, zum Beispiel bei E-Bikes, ein Ladegerät oder ein zweiter Akku. Neben den Marken werden auch die Preise gespeichert.
- Das Bike kann an einen Kunden zu einem bestimmten Preis verkauft werden. Dafür ist das Verkaufsdatum zu speichern.
- Die Kunden werden mit Namen und Adresse gespeichert.
- Bikes können Teil eines Leasingvertrags sein, der von einer Arbeitgeberin oder von einem Arbeitgeber (im Folgenden Arbeitgeber genannt) abgeschlossen werden muss. Ein Arbeitgeber kann beliebig viele Leasingverträge abschließen. Mindestens fünf Bikes sind Teil eines Leasingvertrags. Gespeichert werden die Höhe der monatlichen Leasingrate in Euro, der Beginn und das Ende des Leasingzeitraums sowie der monatliche Versicherungsbeitrag in Euro. Arbeitgeber zeichnen sich durch ihren Firmennamen aus.

Entwickeln und zeichnen Sie ein ER-Diagramm, das die genannten Anforderungen umsetzt.

Hinweis: Das ER-Modell ist in der Form Entitätstypen, Attribute und Beziehungen mit Kardinalitäten in der [min, max]-Notation darzustellen.

(10 BE)

Material 1

Datenblatt und Konfigurationsdatei für das Bike mit der ID 104

BikeID: 104
Anzahl Komponenten: 28
Modell: Manu ATTAIN GTC Carbon
Kategorie: RacingBike

Ausschnitt Komponenten in Einbaureihenfolge

ID	Bezeichnung	Gewicht in g	Preis in €	Hersteller	Gruppe[1]	Name	Abmessung	Detail
r3	Rahmen	1114	1000.50	Manu	Manu A		56 cm	
r7	Gabel	367	299.30	Manu	Manu A		1 1/8“-1 1/4“	
e13	Sattel	187	50.00	MFactory		BergWork	Breite L	
s12	Kurbel	641	222.69	Champions	Form 12			2-fach
e19	Lenker	234	45.00	TeamPro		Evalution	42 cm	
s37	Schaltwerk	232	149.90	Champions	Form 12			2x12-fach
s40	Kette	239	27.70	Champions	Form 12			gekürzt
s10	Bremsgriff r	195	179.00	Champions	Form 12			+Schaltzug
s13	Bremse vorne	163	85.00	Champions	Form 12			Felgen-Br
…								
e65	Hinterrad	1226	249.20	Radelfoxi		TubelessC	2.7 cm	

Hinweis: Die ID der Rahmenteile beginnt mit r, die der Schaltelemente mit s und die der Einzelteile mit e. Es folgt eine Nummer, so dass alle Teile eindeutig zuzuordnen sind.

Aufbau der Konfigurationsdatei

```
r3,Rahmen,1114,1000.50,Manu;Manu A,56 cm<LF>
...
e13,Sattel,187,50.0,MFactory;BergWork,Breite L<LF>
s12,Kurbel,641,222.69,Champions;Form 12,2-fach<LF>
...
e65,Hinterrad,1226,249.20,Radelfoxi;TubelessC,2.7cm<LF>
Manu ATTAIN GTC Carbon;RacingBike<LF>
```

Hinweise: Die Parameter `part0` und `part1` sind am Beispiel der ersten Zeile, durch Semikolon getrennt, wie folgt aufgebaut:

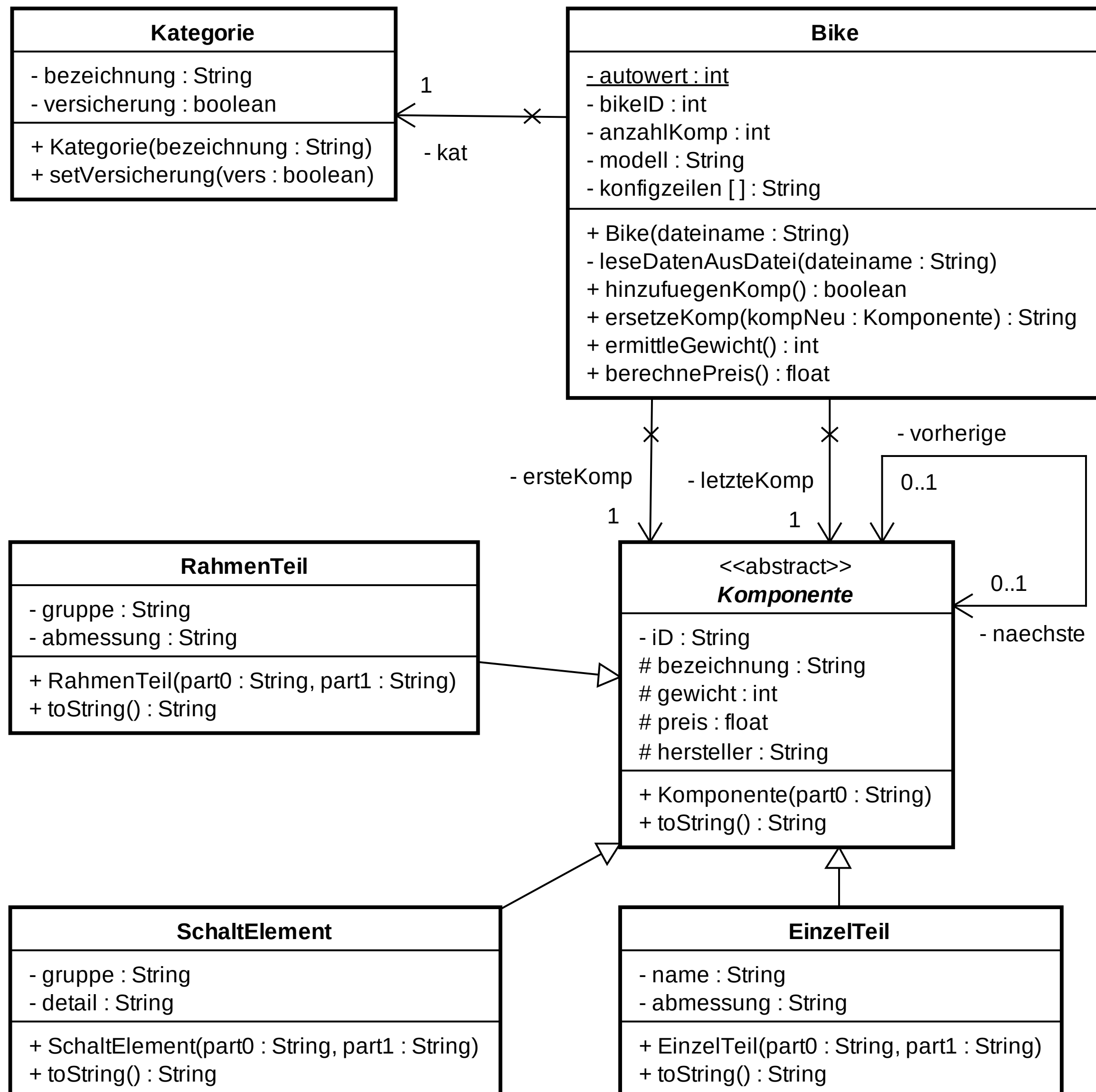
```
part0:     r3,Rahmen,1114,1000.50,Manu
part1:     Manu A,56 cm
```

Die Elemente in der letzten Zeile der Konfigurationsdatei sind die Bezeichnungen für Modell und Kategorie dieses Bikes.

[1] „Gruppe“ ist ein Bike-Fachbegriff, der die Bestandteile des Antriebs und der Bremsen bezeichnet. Zusammengehörige Rahmenteile nennt man Rahmenkit (Rahmenset), aber der Einfachheit halber wird hierfür auch „Gruppe“ verwendet.

Material 2

UML-Klassendiagramm



Hinweise: Die Methode `toString()` der Klasse `Einzelteil` ergibt eine Zeichenkette der Form:
`ID=e19, Lenker wiegt 234 g, kostet 45.00 EUR, Hersteller: TeamPro, Name: Evaluation, 42 cm`

Die private Methode `leseDatenAusDatei()` initialisiert das Attribut `String[] konfigzeilen` der Klasse `Bike` mittels der Konfigurationsdatei (Material 1).

Auf alle Attribute kann mittels get-Methoden zugegriffen werden.

Material 3

Klassendokumentation

Klasse `String`

String
+ equals(str : String) : boolean + split(str : String) : String[ ] + startsWith(str : String) : boolean

`equals(str: String): boolean`

liefert `true`, wenn beide Strings gleich sind, ansonsten `false`.

`split(str: String): String[]`

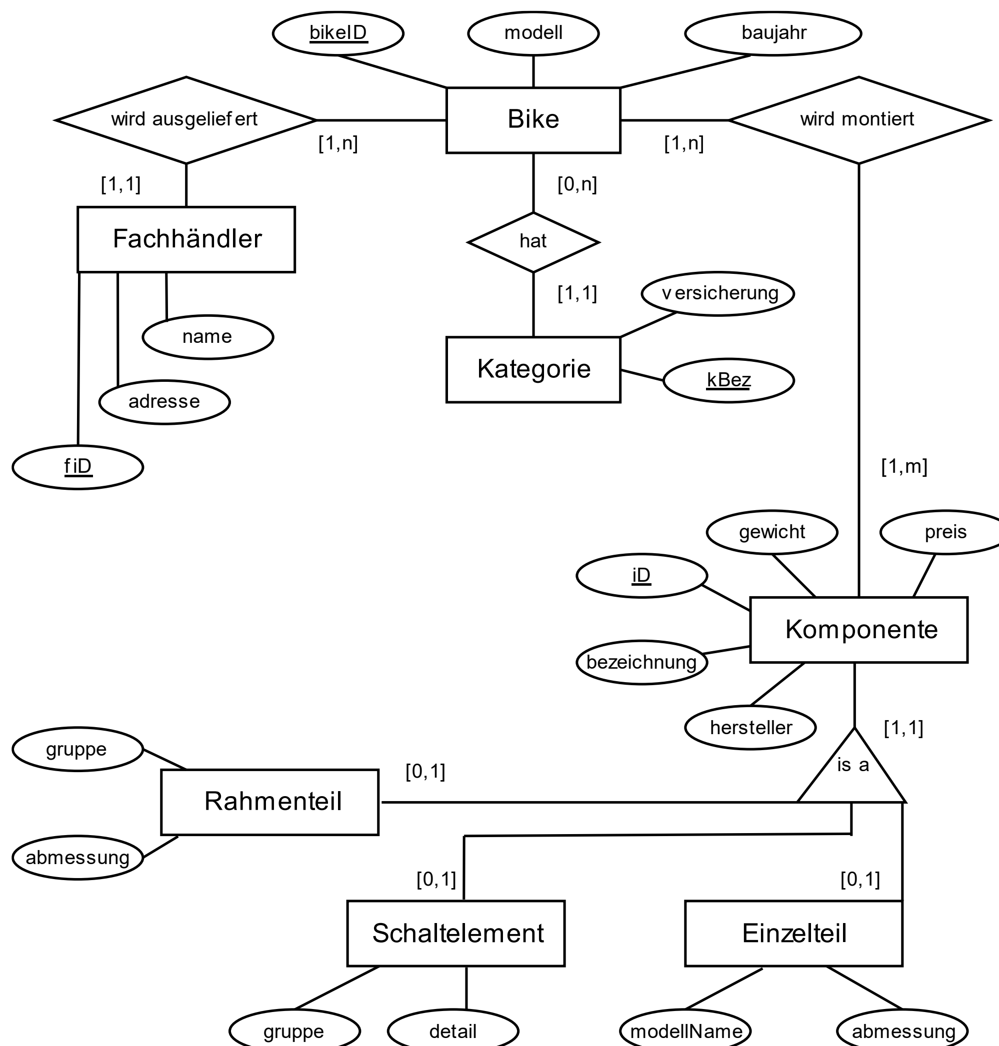
teilt einen String am Trennzeichen `str`. Die Teil-Strings werden in einem Feld zurückgeliefert.

`startsWith(str: String): boolean`

liefert `true`, wenn der String mit `str` beginnt, ansonsten `false`.

Material 4

Entity-Relationship-Modell



Hinweis: Der Datentyp des Attributs `versicherung` ist `boolean`.